

Name: _____

Partners: _____

Python Activity 5: Boolean Expressions

Learning Objectives

Students will be able to:

Content:

- Explain the three types of programming structures
- Explain how conditional operators and logical operators are used in programming
- Use conditional operators with strings and numeric values

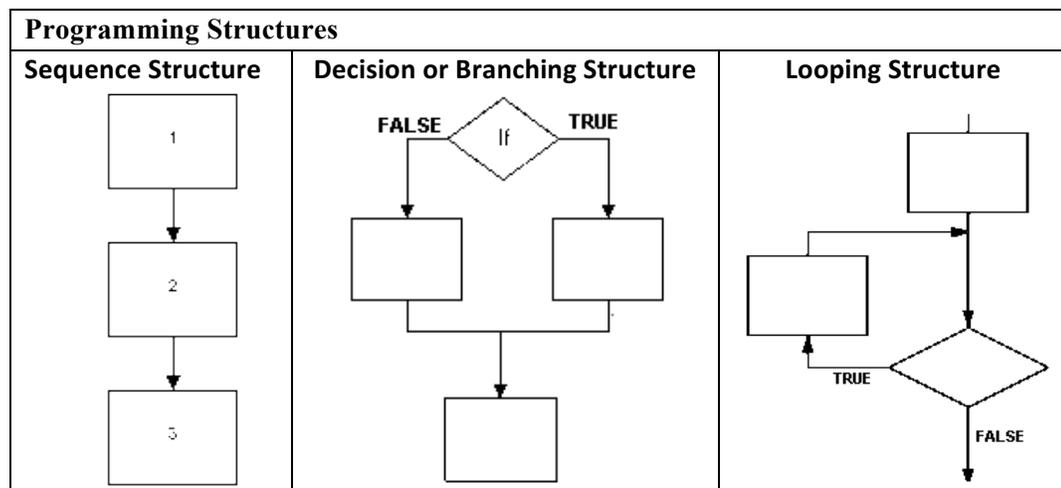
Process:

- Write correct Boolean expressions and compound expressions

Prior Knowledge

- Python concepts from Activities 1-4

Critical Thinking Questions



1. Which structure best describes the types of Python programs you have written so far?

2. Which structure allows the programmer to create code that decides what code is executed?

FYI: Conditional operators, also known as relational operators, are used to compare the relationship between two operands. Expressions whose result can only be **true or false** are known as **Boolean expressions**.

3. State the meaning of each of the following **conditional operators**. If you are not sure of the meaning of any symbol, create some example expressions, type them into the Python *interpreter* (See Figure to the right) and examine the results.

```

=====
>>>
>>> 34 < 56
True

```

- a. < _____ b. > _____
 c. <= _____ d. >= _____
 e. != _____ f. == _____

4. What is the result of each of the following expressions?

Assume: $x = 4$, $y = 5$, and $z = 4$

- a. $x > y$ _____
 b. $x < y$ _____
 c. $x == y$ _____
 d. $x != y$ _____
 e. $x >= z$ _____
 f. $x <= z$ _____
 g. $x + y > 2 * x$ _____
 h. $y * x - z != 4 \% 4 + 16$ _____
 i. $\text{pow}(x,2) == \text{abs}(-16)$ _____

5. What is the result of the following expressions?

Assume: `word1 = "hello"` and `word2 = "good-bye"`

- a. `word1 == word2` _____
 b. `word1 != word2` _____
 c. `word1 < word2` _____
 d. `word1 >= word2` _____

6. How do the conditional operators work when the operands are strings? _____

7. What are the two possible answers for each expression in questions 4 and 5? _____

FYI: We can use **logical operators** to determine logic between conditions (relational expressions).

8. Sometimes you want to test more than one condition to determine which code segment should be executed. You can use the following **logical operators** to create **compound conditions**. Examine each operator and a sample of its use. Provide an explanation of how each operator works.

Operator	Example	Explanation
and	<code>(age >= 17) and (hasLicense == true)</code>	
or	<code>(cost < 20.00) or (shipping == 0.00)</code>	
not	<code>not (credits > 120)</code>	

9. Assume the value of the variable **numBooks** is 40. State the values of each of the Boolean expression.

Expression	Value
(numBooks > 5) and (numBooks < 100)	
(numBooks < 5) or (numBooks > 100)	
not(numBooks * 10 == 100)	

Application Questions: Use the Python Interpreter to check your work

1. Assign a value to **num1** and **num2**. Write a Boolean expression that tests if the value stored in the variable **num1** is equal to the value stored in the variable **num2**.

2. Assign a value to the variables listed in this problem (time, maxTime, cost, and maxCost). Write a Boolean expression that tests if the value stored in the variable **time** is less than the value stored in the variable **maxTime** or if the value stored in the variable **cost** is less than the value stored in the variable **maxCost**

3. Assign a value to **weight** and **cost**. Write a Boolean expression that tests if the value stored in weight is < 10 and the value store in cost is not greater than 20.00

NOTE: COMPLETE **PYTHON ACTIVITY 6: IF-ELSE STATEMENTS** BEFORE MOVING ON TO PYTHON ACTIVITY 7: NESTED IF-ELSE STATEMENTS.

Name: _____

Partner: _____

Python Activity 7: Nested IF-ELSE Statements

Learning Objectives
Students will be able to:

Content:

- Explain the purpose of a nested if-else statement
- Explain how to use Python if-elif structure
- Explain how to test code using Python if-elif structure

Process:

- Write code that includes if-elif statement

Prior Knowledge

- Python concepts from Activities 1-6

Critical Thinking Questions:

1. Closely examine the Python program below.

Python Program

```
def heightMessage(height):  
    female_ht = 162.9 # average US female height in centimeters  
    male_ht = 176.4  
  
    if height > male_ht:  
        print("You're taller than the average American man.")  
    else:  
        if height >= female_ht:  
            print("You have the height of the average American woman, or taller.")  
        else:  
            print("You're not taller than the average.")  
  
def main():  
    heightMessage(float(input("What is your height in cm? ")))  
  
main()
```

- a. In the Python code, circle the if/else statement that is **nested** within another if/else statement.
- b. Enter and test the code. List five numbers to test different parts of this program. Indicate what part of the program the number is testing.

Number	Part Tested

2. Enter and execute the following Python program using the same data as you used for #1b.

```
def question2(height):
    female_ht = 162.9 # average US female height in centimeters
    male_ht = 176.4

    if height > male_ht:
        print("You're taller than the average American man.")
    elif height >= female_ht:
        print("You have the height of the average American woman, or taller.")
    else:
        print("You're not taller than the average.")

def main():
    heightMessage(float(input("What is your height in cm? ")))

main()
```

- a. How does the output for this program compare with the output for the previous program?

- b. What new **keyword** is used in this program? _____
- c. Notice the syntax of this program compared to the previous program. Which program contains simpler indentation? _____

FYI: **elif** is the Python keyword that represents **else if** and allows you to test for one of several options. As soon as one of the tests is true, the rest are ignored.

- d. You can use **elif** as many times as you need to. Suppose you wanted to add the comment “Good!” for grades that are between 80 and 89. Where would you add it? Write the code for this additional choice.

- e. Does it make a difference where the additional **elif** clause is placed? _____
- f. When is the code associated with the **else** statement executed?

g. Change the program so that it prints the following messages. Write the code for the revised program below.

Greater than or equal to 176.4	“Taller than avg man”
Greater than or equal to 162.9 but less than 176.4	“Taller than avg woman
Greater than or equal to 176.4/2 but less than 162.9	“Taller than half the avg men’s height”
Greater than or equal to 162.9/2 but less than 176.4/2	“Taller than half the avg woman’s height”
Less than 162.9/2	“Not taller than average”

h. Make a final change to the program so that it prints an error message if the height entered is greater than 250 or less than 0. Write the additional code below and draw a line to the program above to indicate where you would place this code.

3. Is the use of the **else** statement mandatory when creating an **if/elif** statement? Provide an example to support your answer.

Application Questions: Use the Python Interpreter to check your work

1. Write an if/elif statement that assigns a value to the variable **bonus** depending on the amount of sales. Assume the amount of the sales is stored in a variable called **sales**.

Sales	Bonus
>= 100,000	10,000
>= 75,000	5,000
>= 50,000	2,500
>= 25,000	1,000
